# Assignment 2

Name – Sati, Ankit Date – 2/20/2022

Section – 001

SID – as14128

Total in points (Maximum 100 points)–

Professors Comments –

Affirmation of Independent Effort – Ankit Sati

**Step 1 - Making an Azure Student account.**

* Post setting up the Aws account there are only a small key difference in the Microsoft Azure account.
* We have to setup the storage and server instances as done in the amazon accounts.

1. Before that we need to take care of a few dependencies.

* Python 3.7 (As this will be used in the future for the ArchNav)

Text

Description automatically generated

1. Post this we need to install the JupyterLabs and set the appropriate path.

Text

Description automatically generated

1. We should have the Docker on our machine and it should be up and running.Text

   Description automatically generated
2. Post this, as mentioned in the assignment, we need to run the command (docker run -i -t -p 8888:8888 dbgannon/tutorial).

Text

Description automatically generated

1. Finally, just for the deployment purposes we need to get the kubectl instance up and running with the docker daemons already started.

* Now lets get started with setting up the Azure account.

Subscription type – Azure account

Payment – Pay as you go

Account information.

Subscription ID

5c3b1518-756b-449a-a9cb-424eb50a0f71

Directory

nyu.edu (nyu.edu)

Status

Disabled

Parent management group

7b331012-87a1-4a16-8b0f-a4605b1f3d7f

**Step 2 and 3 – SSH**

SSH pair key

**resource group as14128**

**Connection Key - HostName=as141286.azure-devices.net;DeviceId=myEdgeDevice;SharedAccessKey=DUQ43decm2Rqt8D5u01ZEVDDZ52gq7elK8qdiUKkwBs=**

Text

Description automatically generated

**The CSV file-** https://github.com/domoritz/random-csv

**Step 4 – Creating a storage account with BLOB and tables.**

**Creating the BLOB containers to import the CSV file.**

* First step is to create the file in the blob containers.
* In my case I used the same containers that I had on docker and imported the CSV file onto them.
* Post this data was ready to be exported to Azure.

**Creating the BLOB containers to import the CSV file.**

* Setting up the BLOB in a new storage account.
* In my case I had made the cheapest account that was possible and hence uploaded a very small csv file to capture the data in the BLOB containers.

Graphical user interface, application, Teams

Description automatically generated- Message of successdull creation

Graphical user interface, text, application

Description automatically generated

BLOB Container properties.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Step 5 and 6– Creating a table with the URL links to each of the BLOB and visualizing the data to azure.**

**https://docs.microsoft.com/en-us/azure/cosmos-db/graph/graph-visualization-partners**

Post this we need to push the same amount of files as well as the structure of the BLOB containers should be similar to those in which the CSV files were stored locally.

Note – I have left the file sharer and queues unperformed as they are not required for this assignment.

Adding the URL to the CSV

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Now, Once the URL to each of the csv file is added, we need to create a new container for a new CSV file.

Once that is done we need to repeat the same steps as above to create and add new URL so that we can reach the same BLOB Containers.

Tables

Graphical user interface, text, application, email

Description automatically generated

Uploading data

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**FINAL PART \_ CLEANING UP THE RESOURCES**

**Final screenshot before deletion**

Text

Description automatically generated

**Post Deletion**

Graphical user interface, text, application

Description automatically generated

**Activity log snippet**

Graphical user interface, text, application, email

Description automatically generated

**Again repeating the same steps on AWS.**

1. **Setting up the AWS account with the required instances.**

Required instances.

* EC2(**Compute server**) – This is basically a **regular server** instance that is used to deploy and the required resources over the VM as per the choices made by the users.

- This is used to deploy the VM.

- Manage resources over those VM’s.

- Finally to migrate services and monitor volumes.

* S3(**Storage utility**) – This is a basic protocol that acts like a storage bucket.

- The prime feature of this protocol is to deal with the data as per service request.

- We need this to store the data in the **data buckets** which are later used to store and move the data across volumes created.

**Initial account setup.**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application

Description automatically generated

1. **Step 2 and 3 – SSH into the instance that we have created.**

Text

Description automatically generated

* Saving the required keys as mentioned in the assignment.

Key.pub identification

Type – RSA 4096

Key – Screenshot attached (Blurred the actual key for privacy)

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* The final build needs to be created on the GUI first.
* Post that we need to setup the EC2 instance and the S3 storage bucket
* Finally, we can **ssh** into the created instance.
* **Screenshot of the final build**

Text

Description automatically generated

**The CSV file-** https://github.com/domoritz/random-csv

**Step 4 – NoSQL DB in AWS**

Graphical user interface, text, application

Description automatically generated

After running the notebook, go to aws console and go to dynamodb from there. Click on tables and under that click on DataTable and click on items tab to view the contents of the table.

Graphical user interface

Description automatically generated

**GCP**

Login to GCP account and go to Storage from the navigation panel.

Graphical user interface, text, application

Description automatically generated

Click on create bucket and give a valid name.

Graphical user interface, text, application, email

Description automatically generated

Once the bucket is created, it should be visible

Once done, go to terminal and run the following command:

*docker run -i -t -p 8888:8888 dbgannon/tutorial*

After this go to local host link and run the python notebook called gcloud.ipynb after making necessary edits related to bucket name and file paths.

Graphical user interface, text, application, email

Description automatically generated

Once the above steps are executed, go to GCP again and go to storage to see the uploaded objects.

Table

Description automatically generated

Check the datastore to have a look at created table

Graphical user interface, application, Teams

Description automatically generated

**PART 2**

Generate ssh key using ssh-key gen and following interactive instructions as follows:

Graphical user interface, text, application

Description automatically generated

Go to azure portal and go to virtual machines from the navigation menu

Graphical user interface, text, application, email

Description automatically generated

Click on Add button and after that click on Virtual machine,

Graphical user interface, text, application

Description automatically generated

Enter necessary details on the creation page and then click on Review + create

After validation is passed, click on create

It will prompt to download the generated key pairs, download it.

Once the download is completed, click on “Go to resource”

Graphical user interface, text, application, chat or text message

Description automatically generated

Notice the public IP address of the VM. In order to ssh to this vm, open the terminal and type the command as shown in figure.

Text, letter

Description automatically generated

Download the **x2go** client software as well the **Putty** client for ssh.Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Start the session, a window will appear as below:

A screenshot of a computer

Description automatically generated with medium confidence

Enter the username and password of the virtual machine in order to login to jupyter hub

Once done, jupyter notebooks should be visible

Graphical user interface, text, application, email

Description automatically generated